PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-200542
(43)Date of publication of application : 04.08.1995

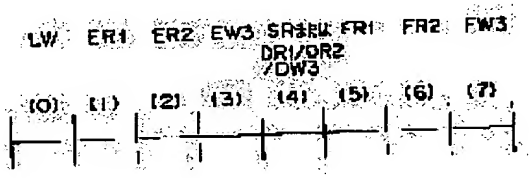
(51)Int.Cl. G06F 17/16
G06F 9/38

(21)Application number : 05-338117
(22)Date of filing : 28.12.1993

(71)Applicant : FUJITSU LTD
(72)Inventor : NAKATANI SHOJI
MOCHIYAMA TAKASHI
KURODA KOJI
KONNO KATSUHIKO
ATSUMI HIROAKI

(54) VECTOR PROCESSOR

(57)Abstract:
PURPOSE: To improve the performance of a vector processor which successively calculate a series of data stored in a main storage device after supplying them to an arithmetic pipeline by performing the arithmetic processing without deteriorating the overall computing efficiency for the arithmetic pipeline of small throughput and by separating the converging period of a vector macroinstruction from the outrun prevention control in regard of the sum total, the retrieval, etc.
CONSTITUTION: If a pipeline that has small throughput compared with other pipelines is detected, a bank management part prescribes the access timing to a vector register of the pipeline. Under such conditions, the pipeline of small throughput uses the timing (SR) set for a memory access pipeline as the access timing (DR1/DR2/DW3) set to the vector register. In addition, a changing part is added on a transmission line of an outrun prevention signal. Thus the outrun prevention signal is variable by a notification signal which notifies a converging period of the relevant instruction.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平7－200542

(43)公開日　平成７年(1995)８月４日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　17/16				
9/38	310　Ｊ			
			Ｇ０６Ｆ　15/347	Ｈ

審査請求　未請求　請求項の数５　ＯＬ　（全 21 頁）

(21)出願番号　特願平5－338117

(22)出願日　平成５年(1993)12月28日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地

(72)発明者　中谷　彰二
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　持山　貴司
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　黒田　浩二
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74)代理人　弁理士　真田　有

最終頁に続く

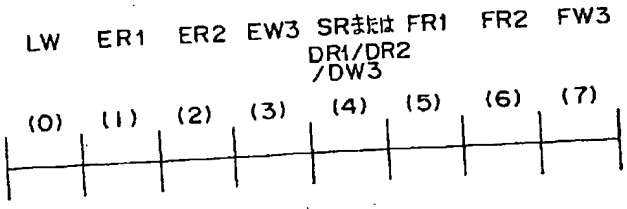
(54)【発明の名称】　ベクトル処理装置

(57)【要約】

【目的】　本発明は、主記憶装置上の一連のデータを順次演算パイプラインに入力して演算するベクトル処理装置に関し、スループットの少ない演算パイプラインについて全体の演算効率を落とすことなく演算処理可能として、演算スループットの向上をはかるとともに、総和，検索等のベクトルマクロ命令の収束期間を追越し防止制御から切り離すようにして性能の向上をはかることを目的とする。

【構成】　他のパイプラインに比べ低スループットのパイプラインが１つでもある場合、バンク管理部によりパイプラインのベクトルレジスタへのアクセスタイミングを規定する際、低スループットのパイプラインがベクトルレジスタをアクセスするタイミング(DR1/DR2/DW3)として、メモリアクセスパイプライン用のタイミング(SR)を用いるように構成するほか、追越し防止信号の伝送路上に変更部を追加し、該当命令の収束期間の通知信号により信号を変更可能に構成する。

第１の発明の原理説明図



1

【特許請求の範囲】

【請求項１】　インターリーブされた複数のバンク単位に複数のエレメント・データを記憶するベクトル・レジスタ（１－０，…，１－ｎ）と、

該ベクトル・レジスタ（１－０，…，１－ｎ）の各エレメント・データをアクセスする複数の演算パイプライン（５－０，…，５－ｍ）および１つまたは複数のメモリ・アクセス・パイプライン（２）と、

前記の演算パイプライン（５－０，…，５－ｍ）およびメモリ・アクセス・パイプライン（２）が各バンク単位をアクセスできるタイミングを示すバンクスロットを管理するバンク管理部（７）とを有し、

前記の演算パイプライン（５－０，…，５－ｍ）およびメモリ・アクセス・パイプライン（２）が該ベクトル・レジスタ（１－０，…，１－ｎ）の各バンク単位を順次アクセスして各エレメント・データを処理するベクトル処理装置において、

前記複数の演算パイプライン（５－０，…，５－ｍ）の中に、他の演算パイプラインに比べて演算スループットの低い演算パイプライン（１５）を少なくとも１つ有し、

該バンク管理部（７）により、前記の演算パイプライン（５－０，…，５－ｍ）およびメモリ・アクセス・パイプライン（２）が該ベクトル・レジスタ（１－０，…，１－ｎ）をアクセスするための各アクセス・タイミングを規定する際に、前記複数の演算パイプライン（５－０，…，５－ｍ）のうちの演算スループットの低い演算パイプライン（１５）が該ベクトル・レジスタ（１－０，…，１－ｎ）をアクセスするタイミングは、前記メモリ・アクセス・パイプライン（２）として割り付けられたタイミングを用いることを特徴とする、ベクトル処理装置。

【請求項２】　前記複数の演算パイプライン（５－０，…，５－ｍ）のうちの演算スループットの低い演算パイプライン（１５）が該ベクトル・レジスタ（１－０，…，１－ｎ）をアクセスするタイミングは、前記メモリ・アクセス・パイプライン（２）のうちのストア・パイプライン（２Ｂ）が該ベクトル・レジスタ（１－０，…，１－ｎ）から主記憶部（２０）へストア動作する読み出しタイミングのバンクスロットに対して割り当てることを特徴とする、請求項１記載のベクトル処理装置。

【請求項３】　インターリーブされた複数のバンク単位に複数のエレメント・データを記憶するベクトル・レジスタ（２１）と、

該ベクトル・レジスタ（２１）上のデータを入力オペランドとするか、もしくは、演算結果を該ベクトル・レジスタ（２１）に書き込む１つまたは複数の演算パイプライン（２２）と、

主記憶部（２４）から該ベクトル・レジスタ（２１）へデータを転送する１つまたは複数のロード・パイプライン（２３）とを有し、

該ロード・パイプライン（２３）から該ベクトル・レジスタ（２１）へデータを転送する命令の実行中に、該ロード・パイプライン（２３）が該ベクトル・レジスタ（２１）に書き込んだデータを入力オペランドとする後続の演算命令を演算パイプライン（２２）が実行する場合、命令の実行順序を保証するために、ロード・パイプライン（２３）の実行を後続の演算パイプライン（２２）の処理が追い越す条件を検出した時に、全ての演算パイプライン（２２）の実行を一時中断する追越し防止制御部（２５）をそなえたベクトル処理装置において、該ベクトル・レジスタ（２１）からのデータ供給を受けるリード処理期間と、該リード処理期間後に結果をまとめ上げる収束期間とを必要とするベクトル命令について、該当ベクトル命令の収束処理を実行中の演算パイプライン（２２）に対する追越し防止制御を行なわないように、該追越し防止制御部（２５）から出力される追越し防止制御信号を変更する変更部（２６）がそなえられていることを特徴とする、ベクトル処理装置。

【請求項４】　該変更部（２６）が、該演算パイプライン（２５）にそなえられていることを特徴とする、請求項３記載のベクトル処理装置。

【請求項５】　該当ベクトル命令の収束処理を実行中の演算パイプライン（２２）が、基本演算器と収束を処理する付加演算器とを有する構成のもので、収束処理を該付加演算器により実行し、収束処理中、該基本演算器により後続の他の演算命令を実行できるものである場合には、収束処理中、該変更部（２６）が、該付加演算器に対してのみ追越し防止制御を行なわないように、該追越し防止制御信号を変更することを特徴とする、請求項３または４に記載のベクトル処理装置。

【発明の詳細な説明】

【０００１】（目次）

【０００２】

【産業上の利用分野】本発明は、主記憶装置に格納されている一連のデータを順次演算パイプラインに入力して演算するベクトル処理装置に関する。

【０００３】

【従来の技術】一般に、ベクトル処理装置においては、例えば、ベクトルＢに属するエレメント・データb_0, b_1,…および／またはベクトルＣに属するエレメント・デ

ータ$c_0, c_1, \cdots$をパイプライン処理によって演算し、その結果得られた$a_0, a_1, \cdots$をベクトルＡに属するエレメント・データとして抽出することが行なわれる。この場合、主記憶装置から置換ロードして演算パイプラインに入力したり、演算パイプラインから主記憶装置に置換ストアしたりすることは、主記憶装置のアクセス速度に制限され、処理速度が遅くなってしまう。

【０００４】このため、通常、主記憶装置と演算パイプラインとの間に、複数個のバンク単位により構成されインターリーブ構造をもつベクトル・レジスタをそなえることが行なわれている。このベクトル・レジスタは、１つのベクトル・レジスタに属するエレメント・データの例えば第ｉ番目のデータと第（ｉ＋１）番目のデータとが互いに異なるバンク単位に格納されるように構成され、各バンク単位の読出出力が互いに異なるパスを介して演算パイプラインに供給されるように構成されるとともに、演算パイプラインから得られた演算結果を互いに異なるパスを介して各バンク単位に書き込むように構成されている。

【０００５】この演算パイプラインが複数ある場合、演算パイプラインからの要請によってベクトル・レジスタへ同時に並列的にアクセスすることが可能な数は、例えば８バンク単位にインターリーブされているとすれば、８個まで可能である。従って、各演算パイプラインからベクトル・レジスタへのアクセスにおいて、同時に同一のバンク単位にアクセスしないように、また、各バンク単位ないし各演算パイプラインが効率よく動作するように、ベクトル・レジスタへのアクセス・タイミングを管理することが極めて重要となる。

【０００６】ベクトル・レジスタへの各エレメント・データの格納に際し、例えば上述のエレメント・データ$b_0, b_1, \cdots$，$c_0, c_1, \cdots$については同じエレメント番号のデータが演算される関係から、同じタイミングで読出を行なうと好都合であるため、可能な限り異なったバンク単位に位置するように格納されている。従って、ベクトル・レジスタへアクセスするためのハードウェアでは、各エレメント・データの最初の格納アドレス（例えば第０番目のエレメントに対するバンク情報）について記憶する手段が必要であるとともに、同時に同一のバンク単位へアクセスしないような作用が必要となり、アクセス制御のためのハードウェアが複雑になってしまう。

【０００７】そこで、従来より、インターリーブされた複数のバンク単位に複数のエレメント・データを記憶するベクトル・レジスタと、このベクトルレジスタの各エレメント・データをアクセスする演算パイプラインおよびメモリ・アクセス・パイプラインと、これらのパイプラインが各バンク単位をアクセスできるタイミングを示すバンクスロットを管理するバンク管理部とを有するベクトル処理装置において、各パイプラインを起動する時点で、バンク管理部からバンクスロット信号を送出することにより、各パイプラインのアクセスタイミングを規定することが行なわれている（例えば特開昭５７－３１０７９号公報参照）。

【０００８】このような従来のベクトル処理装置におけるバンクスロットのタイミング例について、図１９を参照しながら説明する。この図１９において、Ｂ０～Ｂ７はそれぞれ８個のバンク単位を示し、ＬＷは、メモリ・アクセス・パイプラインのうちのロード・パイプラインが主記憶装置（メモリ）からベクトル・レジスタにデータを書き込むタイミングを示し、ＳＲは、メモリ・アクセス・パイプラインのうちのストア・パイプラインがベクトル・レジスタから主記憶装置へ読み出すタイミングを示す。

【０００９】また、Ｅ，Ｆはそれぞれ例えば２種類の演算パイプライン（乗算もしくは加算パイプライン）が動作するバンクスロットの区別を示し、各Ｅ，Ｆに付されたＲ，Ｗはそれぞれ各オペランドによるベクトル・レジスタから各演算パイプラインへの読出(READ)，各演算パイプラインからベクトル・レジスタへの書込（WRITE)に対応するものであり、Ｒ，Ｗに付された数字１，２，３はそれぞれ各オペランドＯＰ１，ＯＰ２，ＯＰ３の番号に対応している（ＯＰ１＊ＯＰ２⇒ＯＰ３；＊は演算子）。

【００１０】さらに、（０）～（７）はカウント値を示している。この図１９に示すように、従来、演算パイプラインのように固定パイプライン長として割り当てるバンクスロットには、ＥＲ１，ＥＲ２，ＥＷ３が割り当てられ、メモリ・アクセス・パイプラインのようにパイプライン長が不定のパイプラインには、ＬＷまたはＳＲが割り当てられている。

【００１１】一方、ベクトル処理装置では、処理を高速化するために、１つの命令の実行完了を待たずに後続の命令の実行を開始するが、処理スループットの異なるパイプライン間で結果オペランドと入力オペランドとに依存関係がある場合には、追越し防止制御を行なう必要がある。複数のパイプライン間のデータリンクの状況を正確に確認して適切な追越し防止制御を行なうには、物量的なインパクトが大きく、遅延時間も大きくなるので、現実的ではない。

【００１２】そのため、適切さは欠くが簡便な手法が従来より使用されている。その１つが、少なくとも１つの演算パイプラインがリンク状態にあるときには、追越し禁止制御を行なう際に、対象とする演算パイプラインのみでなく、全演算パイプラインに対して追越し禁止制御を適用するという手法である。つまり、従来の追越し制御防止部は、ロード命令が他のパイプラインとリンク動作（チェイニングともいう）を開始するという情報を命令発信／管理部から受け取ると、そのロード命令が完了するまで、追越し防止制御を活性化する。追越し防止制御部は、リンク動作しているロード・パイプラインが、

メモリ・バス・コンフリクト等の要因で、所定のスループットを下回るおそれがあることを検出すると、追越し防止制御信号を立ち上げることにより、演算パイプライン，ストア・パイプラインの実行を中断させる。

【００１３】

【発明が解決しようとする課題】ところで、図１９に示すようにバンクスロットのタイミングを設定する場合、固定長パイプラインの中でも、乗算パイプラインや加算パイプラインは、各サイクル、演算結果を得ることが可能であるが、割算パイプラインのように各サイクル毎に数ビットの結果しか得られないものもある（乗算／加算時に比べ、１／３以下のパイプラインで、例えば１／７のスループット）。

【００１４】このようなスループットの少ない演算パイプライン、つまり読出あるいは書込に要するバンクの使用時間が少ない演算パイプラインが、バンクスロット（ＥＲ１，ＥＲ２，ＥＷ３，ＦＲ１，ＦＲ２，ＦＷ３）を占有することは、全体の演算効率を落とし、ベクトル処理装置全体のスループットの低下を招くという課題があった。

【００１５】また、従来の追越し防止制御方式では、他のパイプラインの動作フェイズには関与しないため、過剰にパイプラインを止めてしまうことがある。例えば、総和命令，検索命令，抽出命令のように最終的に１つの結果を求める演算では、演算途中で発生する中間的な結果をまとめ上げるための収束期間が存在するが、このような収束期間は、原理的に前述のリンク動作とは独立に動作することが可能であるにもかかわらず、従来の追越し防止制御では、その追越し防止制御信号による処理停止のために収束処理までも停止してしまい、処理速度の低下を招くなどの課題もあった。

【００１６】本発明はこのような課題に鑑み創案されたもので、本発明の第１の目的は、スループットの少ない演算パイプラインについて、全体の演算効率を落とすことなく演算処理を行なえるようにして、演算スループットの向上をはかったベクトル処理装置を提供することである。また、本発明の第２の目的は、演算パイプラインが総和，検索等のベクトルマクロ命令の収束処理のシーケンス実行中である場合には、追越し防止制御を行なわないように制御することにより、処理速度の改善をはかったベクトル処理装置を提供することである。

【００１７】

【課題を解決するための手段】図１は第１の発明の原理説明図である。第１の発明のベクトル処理装置も、基本的には従来のベクトル処理装置と同様に、インターリーブされた複数のバンク単位に複数のエレメント・データを記憶するベクトル・レジスタと、このベクトル・レジスタの各エレメント・データをアクセスする複数の演算パイプラインおよび１つまたは複数のメモリ・アクセス・パイプラインと、これらのパイプラインが各バンク単位をアクセスできるタイミングを示すバンクスロットを管理するバンク管理部とから構成され、演算パイプラインおよびメモリ・アクセス・パイプラインがベクトル・レジスタの各バンク単位を順次アクセスして各エレメント・データが処理されるようになっている。

【００１８】そして、第１の発明では、複数の演算パイプラインの中に、他の演算パイプラインに比べて演算スループットの低い演算パイプラインを少なくとも１つ有する場合、図１に示すように、バンク管理部により、複数の演算パイプラインのうちの演算スループットの低い演算パイプラインがベクトル・レジスタをアクセスするタイミング（ＤＲ１，ＤＲ２，ＤＷ３）は、メモリ・アクセス・パイプラインとして割り付けられたタイミング、特にメモリ・アクセス・パイプラインのうちのストア・パイプラインがベクトル・レジスタから主記憶部へストア動作する読み出しタイミング（ＳＲ）のバンクスロットに対して割り当てられている（請求項１，２）。なお、図１中の各符号は、図１９により前述したものと同様であるが、Ｄは、演算スループットの低い演算パイプラインが動作するバンクスロットの区別を示している。

【００１９】また、図２は第２の発明の原理ブロック図で、この図２において、２１はインターリーブされた複数のバンク単位に複数のエレメント・データを記憶するベクトル・レジスタ、２２はベクトル・レジスタ２１上のデータを入力オペランドとするかもしくは演算結果をベクトル・レジスタ２１に書き込む１つまたは複数の演算パイプライン、２３は主記憶部２４からベクトル・レジスタ２１へデータを転送する１つまたは複数のロード・パイプラインである。

【００２０】また、２５は追越し防止制御部で、この追越し防止制御部２５は、ロード・パイプライン２３からベクトル・レジスタ２１へデータを転送する命令の実行中に、ロード・パイプライン２３がベクトル・レジスタ２１に書き込んだデータを入力オペランドとする後続の演算命令を演算パイプライン２２が実行する場合、命令の実行順序を保証するために、ロード・パイプライン２３の実行を後続の演算パイプライン２２の処理が追い越す条件を検出した時に、全ての演算パイプライン２２の実行を一時中断するものである。

【００２１】そして、第２の発明では、変更部２６が新たにそなえられている。この変更部２６は、ベクトル・レジスタ２１からのデータ供給を受けるリード処理期間と、リード処理期間後に結果をまとめ上げる収束期間とを必要とするベクトル命令については、該当ベクトル命令の収束処理を実行中の演算パイプライン２２に対する追越し防止制御を行なわないように、追越し防止制御部２５から出力される追越し防止制御信号を変更するものである（請求項３）。

【００２２】なお、この変更部２６は、演算パイプライ

ン２２にそなえてもよい（請求項４）。また、該当ベクトル命令の収束処理を実行中の演算パイプライン２２が、基本演算器と収束を処理する付加演算器とを有する構成のもので、収束処理を該付加演算器により実行し、収束処理中、基本演算器により後続の他の演算命令を実行できるものである場合には、収束処理中、変更部２６が、付加演算器に対してのみ追越し防止制御を行なわないように、追越し防止制御信号を変更してもよい（請求項５）。

【００２３】

【作用】上述した第１の発明のベクトル処理装置（請求項１，２）では、演算パイプラインのように固定長のパイプラインにおいても、特にスループットの少ないパイプラインを、メモリ・アクセス・パイプラインのような１つのバンクスロットしか使用しないパイプラインと共用することにより、演算パイプラインをオーバラップさせて実行している。

【００２４】また、上述した第２の発明のベクトル処理装置（請求項３）では、追越し防止制御部２５から追越し防止制御信号が出力された際に、演算パイプライン２２が収束処理のシーケンスを実行中で、収束期間条件が成立している間は、変更部２６により追越し防止制御部２５からの追越し防止制御信号が変更され、収束処理中の演算パイプライン２２に対する追越し防止制御が禁止される。

【００２５】なお、この変更部２６を演算パイプライン２２にそなえた場合には、収束処理を実行中の演算パイプラインが、追越し防止制御部２５からの追越し防止制御のための信号を無視する形で、演算パイプライン２２に対する追越し防止制御部２５による追越し防止制御が禁止される（請求項４）。また、収束処理中の演算パイプライン２２が基本演算器と収束処理用の付加演算器とをもつものである場合には、変更部２６により追越し防止制御部２５からの追越し防止制御信号を変更することで、付加演算器に対してのみ追越し防止制御が禁止される（請求項５）。

【００２６】

【実施例】以下、図面を参照して本発明の実施例を説明する。

（ａ）第１実施例の説明

図３は本発明の第１実施例としてのベクトル処理装置を示すブロック図で、この図３において、１－０，１－１，…，１－ｎはそれぞれベクトル・レジスタで、各ベクトル・レジスタ１－０，１－１，…，１－ｎは、それぞれ、インターリーブされたバンク単位Ｂ０，Ｂ１，…，Ｂ７（本実施例では８バンク単位の場合を示す）に複数のエレメント・データを記憶するものである。

【００２７】２は主記憶部２０とベクトル・レジスタ１－０，１－１，…，１－ｎとの間において各エレメント・データを高速にロードないしストアすべくパイプライン構成されたメモリ・アクセス・パイプラインで、ロード・パイプライン２Ａおよびストア・パイプライン２Ｂを有している。ここで、ロード・パイプライン２Ａは、主記憶部２０からのエレメント・データをベクトル・レジスタ１－０，１－１，…，１－ｎへロードするためのものであり、ストア・パイプライン２Ｂは、ベクトル・レジスタ１－０，１－１，…，１－ｎに格納されたエレメント・データを主記憶部２０へストアするためのものである。

【００２８】３Ａはメモリ・アクセス・パイプライン２のロード・パイプライン２Ａからのエレメント・データをベクトル・レジスタ１－０，１－１，…，１－ｎにおける各バンク単位Ｂ０～Ｂ７に書き込むための書込レジスタ、３Ｂ－０，３Ｂ－１，…，３Ｂ－ｍはそれぞれ後述する演算パイプライン５－０，５－１，…，５－ｍからのエレメント・データ（演算結果）をベクトル・レジスタ１－０，１－１，…，１－ｎにおける各バンク単位Ｂ０～Ｂ７に書き込むための書込レジスタである。

【００２９】４Ａはベクトル・レジスタ１－０，１－１，…，１－ｎに格納されたエレメント・データをメモリ・アクセス・パイプライン２のストア・パイプライン２Ｂへ読み出すための読出レジスタである。また、４Ｂ－０，４Ｃ－０；４Ｂ－１，４Ｃ－１；…；４Ｂ－ｍ，４Ｃ－ｍはそれぞれ後述する演算パイプライン５－０，５－１，…，５－ｍ毎にそなえられた一対の読出レジスタで、各対の読出レジスタ４Ｂ－０，４Ｃ－０；４Ｂ－１，４Ｃ－１；…；４Ｂ－ｍ，４Ｃ－ｍは、それぞれ、演算対象となる一対のエレメント・データを演算パイプライン５－０，５－１，…，５－ｍに入力すべく、その一対のエレメント・データをベクトル・レジスタ１－０，１－１，…，１－ｎにおける各バンク単位Ｂ０～Ｂ７から読み出すためのものである。

【００３０】５－０，５－１，…，５－ｍは演算パイプラインで、これらの演算パイプライン５－０，５－１，…，５－ｍは、それぞれ、一対の読出レジスタ４Ｂ－０，４Ｃ－０；４Ｂ－１，４Ｃ－１；…；４Ｂ－ｍ，４Ｃ－ｍを介して入力されたエレメント・データに対して四則演算等の演算処理を施し、その演算結果（エレメント・データ）を出力するものである。

【００３１】そして、６は各種のベクトル演算命令を出力する命令制御部、７は命令制御部６からの命令を受けて動作するバンク管理部で、このバンク管理部７は、メモリ・アクセス・パイプライン２および演算パイプライン５－０，５－１，…，５－ｍがベクトル・レジスタ１－０，１－１，…，１－ｎにおける各バンク単位Ｂ０～Ｂ７をアクセスできるタイミングを示すバンクスロットを管理するもので、各バンク単位Ｂ０～Ｂ７へのアクセス・タイミングを規制するバンクスロット・カウンタ７ａを有している。

【００３２】次に、上述のようなベクトル処理装置の一

般的な動作について説明する。図３に示す本実施例のベクトル処理装置では、各ベクトル・レジスタ１－０，１－１，…，１－ｎは、各バンク単位Ｂ０～Ｂ７にそれぞれ分散するように対応付けられている。そして、各ベクトル・レジスタ１－０，１－１，…，１－ｎに格納されるエレメント・データは、第０番目のデータがバンク単位Ｂ０に記憶され、第１番目のデータがバンク単位Ｂ１に記憶され、第７番目のデータがバンク単位Ｂ７に記憶されるというように各バンク単位に順次記憶され、いわゆるインターリーブした形に格納され、同じナンバのデータが同じバンク単位に位置するように格納される。

【００３３】例えば、ベクトルＢに属するエレメント・データb_0, b_1,…が主記憶部２０からロードされてベクトル・レジスタ１－１内に格納されているものとし、またベクトルＣに属するエレメント・データc_0, c_1,…が同様にベクトル・レジスタ１－２に格納されているものとする。この状態で、例えば、命令制御部６からバンク管理部７に対して、ベクトル加算命令「ＯＰ１〔＃１ＶＲ（ｉ）〕＋ＯＰ２〔＃２ＶＲ（ｉ）〕⇒ＯＰ３〔＃０ＶＲ（ｉ）〕」が与えられたとすると、バンク管理部７により、次のごとく①～⑥の処理（図４参照）が実行される。なお、本実施例では、演算パイプライン（加算パイプライン）５－０が３段のステップ段数をもつものとする（図４参照）。

【００３４】ここで、＃０ＶＲ（ｉ），＃１ＶＲ（ｉ），＃２ＶＲ（ｉ）はそれぞれベクトル・レジスタ１－０，１－１，１－２の各バンク単位Ｂｉに格納されるデータを意味し、上記ベクトル加算命令は、ベクトル・レジスタ１－１に格納された各エレメント・データと、ベクトル・レジスタ１－２に格納された各エレメント・データとを加算し、その加算結果（エレメント・データ）をベクトル・レジスタ１－０に格納する命令となっている。

【００３５】①タイミング・サイクル（バンクスロット・カウンタ７ａによりカウントされるカウント値に対応するもの）Ｔ０，Ｔ１，…において、バンク単位Ｂ０，Ｂ１，…，Ｂ７，…に対して、順次、リード・アクセスが行なわれ、その結果、読出レジスタ４Ｂ－０，４Ｃ－０を介して、エレメント・データb_0, b_1,…およびc_0, c_1,…が、順次、ベクトル・レジスタ１－１，１－２から読み出される。

【００３６】②タイミング・サイクルＴ２において、データb_0とc_0とは演算パイプライン５－０のステップＩに入力される。

③タイミング・サイクルＴ３において、データb_0とc_0とは演算パイプライン５－０のステップIIに入力されると同時に、データb_1とc_1とが演算パイプライン５－０のステップＩに入力される。

【００３７】④タイミング・サイクルＴ４において、データb_0とc_0とは演算パイプライン５－０のステップIII に入力され、データb_1とc_1とは演算パイプライン５－０のステップIIに入力されると同時に、データb_2とc_2とが演算パイプライン５－０のステップＩに入力される。

⑤タイミング・サイクルＴ５において、データb_0とc_0との加算結果であるデータa_0が書込レジスタ３Ｂ－０にセットされる。

【００３８】⑥タイミング・サイクルＴ６において、このデータa_0が、ベクトル・レジスタ１－０のバンク単位Ｂ０に書き込まれる。以下、順次得られるデータa_1, a_2,…が書込レジスタ３Ｂ－０にセットされ、書込レジスタ３Ｂ－０にセットされたデータa_1, a_2,…は、それぞれ、ベクトル・レジスタ１－０のバンク単位Ｂ１，Ｂ２，…，Ｂ７，Ｂ０，…に順次に書き込まれる。

【００３９】ここで、演算パイプライン５－０では、同じナンバ（添字番号）のエレメント・データがステップＩに入力されるように、ベクトルＢ，Ｃに属するエレメント・データの入力側にタイミングを合わせるためのバッファ・レジスタが１段設けられている。このように構成することによって、ベクトルＢ，Ｃに関して加算し、その結果得られたベクトルＡをバンク単位Ｂ０，Ｂ１，…の順にアクセスすることが可能になる。

【００４０】ついで、前記の各バンク単位Ｂ０～Ｂ７へのアクセス制御を簡略化すべく、そのアクセス制御を行なうバンク管理部７について、図５，図６を参照しながら説明する。図５において、１１－１，１１－２，１１－３はメモリ・アクセス・パイプライン２もしくは演算パイプライン５－０，５－１，…，５－ｍがベクトル・レジスタ１－０，１－１，…，１－ｎにアクセスするタイミング・サイクル（以下、バンクスロットという）を記憶する管理レジスタ、１２はバンクスロット割当回路、１３はバンクスロットを記憶しメモリ・アクセス・パイプライン２へ通知する通知レジスタ、１４は起動信号制御部である。

【００４１】バンク管理部７では、各バンク単位Ｂ０～Ｂ７にアクセスするパイプライン装置（メモリ・アクセス・パイプライン２もしくは演算パイプライン５－０，５－１，…，５－ｍ）が同一バンク単位へ同時にアクセスすることのないように、さらには無駄な空き時間を生じない効率のよいアクセスが可能になるアクセス制御を実現するために、バンクスロット・カウンタ７ａ（１個設けられている）を、常時カウントすることによって、バンクスロットと呼ばれるタイミング・サイクルＴ０～Ｔ７を規定し、そのカウンタ出力信号を各パイプライン装置へ通知している。

【００４２】なお、このとき、バンクＢ１をアクセスするタイミングは、バンクＢ０よりも１サイクル遅れた状態であるため、本カウンタ７ａによるカウント値が“１”の時、１サイクル前でバンクＢ０をアクセスしていたパイプラインがバンクＢ１をアクセスする。従っ

て、バンクスロット・カウンタ７ａによるカウント値は、バンクＢ０にアクセスするパイプラインの順を示している。

【００４３】各管理レジスタ１１－１，１１－２，１１－３は、各パイプライン装置が有する各バンク単位にデータ転送するためのチャネル（アクセス要求）に対して、起動する時点において割り当てられるバンクスロット番号（Ｂ１）を記憶する例えば３ビットの記憶素子（実際には記憶内容の有効／無効表示のためにさらに１ビットが必要）で構成されるものであり、パイプライン装置がベクトル・レジスタ１－０，１－１，…，１－ｎにアクセスしている期間、そのチャネル（アクセス要求）に割り当てられたバンクスロット番号を記憶している。

【００４４】バンクスロット割当回路１２は、管理レジスタ１１－１，１１－２，１１－３の出力と、バンクスロット・カウンタ７ａの出力とによって、使用中のバンクスロット番号と現在のバンクスロット番号とを知り、起動のあるパイプライン装置に対して空き時間の最少となるようなバンクスロット番号を割り当てる選択回路である。

【００４５】ここで、命令制御部６が、バンク管理部７に起動信号を与え、例えば、メモリ・アクセス・パイプライン２が主記憶部２０へアクセスして読出データをベクトル・レジスタ１－０へデータ転送するように要求したとする。このとき、バンクスロット割当回路１２は、メモリ・アクセス・パイプライン２を起動する時点において、当該チャネル（アクセス要求）に相当する管理レジスタ１１－１，１１－２，１１－３に、その選択したバンクスロット番号をセット番号として伝え記憶させる。

【００４６】図６は上述の動作を詳細に説明するためのもので、この図６に示すように、メモリ・アクセス・パイプライン２が起動信号によって起動されると、エレメント・データa_0，a_1，…，a_n（主記憶部２０に格納されているものとする）にメモリ・アクセスが開始され、アクセス・タイムt_Aの後に、メモリ・アクセス・パイプライン２のバッファ・レジスタ（図５には図示せず）にその読出内容がロードされる。

【００４７】各パイプライン装置には、予め決められたバンクスロット（図１，図７，図１９参照）が割り当てられているので、各パイプライン装置でバンクスロット・カウンタ７ａの出力を参照し、所望のタイミングになった時点で、バッファ・レジスタに一時記憶していたエレメント・データa_0，a_1，…，a_nをベクトル・レジスタ１－０，…へ順次転送する。最後のデータを転送すると、パイプライン終了信号によってリセット信号をバンク管理部７へ送出し、当該転送チャネルに相当する管理レジスタ１１－１または１１－２，１１－３の内容をリセットし無効にする。

【００４８】なお、バッファ・レジスタは、主記憶部２０の読出出力をストローブするタイミングと、バンクスロットによりベクトル・レジスタ１－０，…に書き込むまでの期間を調整する複数個のレジスタである。また、上述した例では、主記憶部２０が他のアクセス装置からの要求によってビジー状態等であることを考慮すると、起動してからベクトル・レジスタ１－０を使用するまでの時間が一定でない。即ち、上述の例では、パイプライン長が不定な装置とすることができ、演算パイプライン５－０，５－１，…，５－ｍは、ステップ数（即ちパイプライン長）が固定であり、主記憶部２０の状況に影響されず起動されてからベクトル・レジスタ１－０を使用するまでの時間は一定となる。この場合は、図４において説明した通り、演算パイプライン５－０，５－１，…，５－ｍが、ベクトル・レジスタ１－０にアクセスするタイミング関係は固定となっている。

【００４９】このため、パイプライン長が固定の装置において、各チャネル（アクセス要求）間のアクセス・タイミングのずれを認識することにより、図５に示すごとく、起動信号制御部１４において、使用中のバンクスロット番号，バンクスロット・カウンタ７ａの内容および起動信号に基づき、起動信号のタイミングでバンクスロットのタイミングを判断することが可能である。

【００５０】さて、本実施例では、上述のような一般的な動作を行なうベクトル処理装置において、複数の演算パイプライン５－０，５－１，…，５－ｍの中に、他の演算パイプラインに比べて演算スループットの低い演算パイプラインとして、割算パイプライン１５（６個の割算器５ａ～５ｆを有してなるパイプライン）を有する場合に、バンク管理部７は、図７に示すように、バンクスロットを管理している。

【００５１】つまり、バンク管理部７は、各割算パイプライン５ａ～５ｆがベクトル・レジスタ１－０，１－１，…，１－ｎをアクセスするタイミング（ＤＲ１，ＤＲ２，ＤＷ３）を、メモリ・アクセス・パイプライン２のうちのストア・パイプライン２Ｂがベクトル・レジスタ１－０，１－１，…，１－ｎから主記憶部２０へストア動作する読み出しタイミング（ＳＲ）のバンクスロットに対して割り当てている。

【００５２】なお、図７において、各符号は、図１９により前述したものと同様であるが、ＤＲ１，ＤＲ２は割算パイプライン１５における読出オペランドのタイミングを示し、ＤＷ３は割算パイプライン１５における書込オペランドのタイミングを示している。割算パイプライン１５が動作している間は、タイミングを時分割で使用する。また、割算パイプライン１５とストア・パイプライン２Ｂとはいずれか一方のみ動作するように制御する。さらに、Ｅ，Ｆは演算パイプライン５－０，５－１，…，５－ｍのうちの加算パイプラインもしくは乗算パイプラインが動作するバンクスロットの割付タイミン

グを示す。

【００５３】本実施例では、図７に示すように、２種類の演算パイプラインが動作している際に、各バンク単位Ｂ０～Ｂ７で割算パイプライン１５の各割算器５ａ～５ｆは１６タイミング・サイクル毎に到来する、ストア・パイプライン２Ｂがベクトル・レジスタ１－０，１－１，…，１－ｎから主記憶部２０へストア動作する読み出しタイミングＳＲのバンクスロットを共用して、割算パイプライン１５による演算処理を実行している。

【００５４】例えば、割算パイプライン１５内の割算器５ａでは、バンク単位Ｂ０の最初のストア読出タイミングＳＲ（タイミング・サイクルＴ４）と、バンク単位Ｂ１の最初ストア読出タイミングＳＲ（タイミング・サイクルＴ５）とで、ベクトル・レジスタ１－０，１－１，…，１－ｎからのデータの読出（ＤＲ１，ＤＲ２）を行なう。そして、１６タイミング・サイクル後のバンク単位Ｂ１の２周期目のストア読出タイミングＳＲ（タイミング・サイクルＴ５）で、割算結果のベクトル・レジスタ１－０，１－１，…，１－ｎへの書込（ＤＷ３）を行なった後、再び、バンク単位Ｂ２の２周期目のストア読出タイミングＳＲ（タイミング・サイクルＴ６）と、バンク単位Ｂ３の２周期目のストア読出タイミングＳＲ（タイミング・サイクルＴ７）とで、ベクトル・レジスタ１－０，１－１，…，１－ｎからのデータの読出（ＤＲ１，ＤＲ２）を行ない、以下、同様の処理を繰り返す。

【００５５】また、割算パイプライン１５内の割算器５ｂでも、同様に、バンク単位Ｂ３の最初のストア読出タイミングＳＲ（タイミング・サイクルＴ７）と、バンク単位Ｂ４の最初ストア読出タイミングＳＲ（タイミング・サイクルＴ０）とで、ベクトル・レジスタ１－０，１－１，…，１－ｎからのデータの読出（ＤＲ１，ＤＲ２）を行ない、１６タイミング・サイクル後のバンク単位Ｂ４の２周期目のストア読出タイミングＳＲ（タイミング・サイクルＴ０）で、割算結果のベクトル・レジスタ１－０，１－１，…，１－ｎへの書込（ＤＷ３）を行ない、以下、同様の処理を繰り返す。その他の割算器５ｃ，５ｄ，５ｅ，５ｆについても同様に、タイミングの割付を行なって、割算処理が行なわれる。

【００５６】このように、本発明の第１実施例のベクトル処理装置によれば、演算パイプラインのように固定長のパイプラインにおいても、特にスループットの少ないパイプライン、例えば割算パイプライン１５を、メモリ・アクセス・パイプライン２のような１つのバンクスロットしか使用しないパイプラインと共用することにより、演算パイプラインをオーバラップさせて実行し、全体の演算効率を落とすことなく演算処理を行なえ、演算スループットが大幅に向上することになる。

【００５７】また、特に、メモリ・アクセス・パイプライン２の中でも、主記憶部２０から読出を行なうロード・パイプライン２Ａは演算のソースとなるため、結果オペランドを格納するためのストア・パイプライン２Ｂのバンクスロットと共用することにより、ベクトル処理装置全体のスループットを向上させることができるのである。

【００５８】なお、上述の実施例では、スループットの少ないパイプラインが割算パイプラインである場合について説明したが、本発明は、これに限定されるものではなく、スループットの少ないパイプラインがスクエア・ルート（２乗根）演算パイプライン等である場合にも同様に適用され、上述した実施例と同様の作用効果が得られる。

【００５９】（ｂ）第２実施例の説明図８は本発明の第２実施例としてのベクトル処理装置を示すブロック図で、この図８において、２１はインターリーブされた複数のバンク単位に複数のエレメント・データを記憶するベクトル・レジスタ（ＶＲ）、２２－１，２２－２，…，２２－ｎはそれぞれベクトル・レジスタ２１上のデータを入力オペランドとするかもしくは演算結果をベクトル・レジスタ２１に書き込む演算パイプラインである。

【００６０】また、２３－１，２３－２，…，２３－ｍはベクトル・レジスタ２１と主記憶部２４との間において各エレメント・データを高速にロードないしストアすべくパイプライン構成されたメモリ・アクセス・パイプラインで、各メモリ・アクセス・パイプライン２３－１，２３－２，…，２３－ｍは、ロード・パイプライン２３Ａおよびストア・パイプライン２３Ｂとしての機能を有している。ここで、ロード・パイプライン２３Ａは、主記憶部２４からベクトル・レジスタ２１へデータを転送（ロード）するためのものであり、ストア・パイプライン２３Ｂは、ベクトル・レジスタ２１に格納されたデータを主記憶部２４へ転送（ストア）するためのものである。

【００６１】なお、図８で示した各メモリ・アクセス・パイプライン２３－１，２３－２，…，２３－ｍでは、ロード・パイプライン２３Ａとストア・パイプライン２３Ｂとがペアとなってそなえられているが、いずれか一方のみを有する構成としてもよい。また、メモリ・アクセス・パイプライン２３－１，２３－２，…，２３－ｍのいずれかがロード・パイプライン２３Ａもしくはストア・パイプライン２３Ｂの一方のみ有する構成としてもよい。

【００６２】２５は追越し防止制御部で、この追越し防止制御部２５は、メモリ・アクセス・パイプライン２３－１，２３－２，…，２３－ｍにおけるロード・パイプライン２３Ａからベクトル・レジスタ２１へデータを転送する命令の実行中に、ロード・パイプライン２３Ａがベクトル・レジスタ２１に書き込んだデータを入力オペランドとする後続の演算命令を、演算パイプライン２２

15

－１，２２－２，…，２２－ｎのいずれかが実行する場合、命令の実行順序を保証するために、ロード・パイプライン２３Ａの実行を後続の演算パイプラインの処理が追い越す条件を検出した時に、全ての演算パイプライン２２－１，２２－２，…，２２－ｎの実行を一時中断すべく、例えば、処理の一時中断を要求している間は“１”に立ち上がり、処理を続行可能な時には“０”になる追越し防止制御信号を出力するものである。

【００６３】２７は各演算パイプライン２２－１，２２－２，…，２２－ｎや各メモリ・アクセス・パイプライン２３－１，２３－２，…，２３－ｍに対する命令の発信と進行状況とを管理する命令発信／管理部で、この命令発信／管理部２７は、命令を発信するときに、その命令の発信以前に実行を開始され未だその実行を完了していない命令との間にオペランド・レジスタのリンクがあるかどうかをチェックし、リンクがあれば、追越し防止制御部２５にリンク条件が成立したことをリンク情報として通知する機能を有している。

【００６４】そして、第２実施例では、図９に示すような追越し防止制御信号変更部２６Ａおよび収束期間通知部２６Ｂが、各演算パイプライン２２－１，２２－２，…，２２－ｎ毎にそなえられている。この追越し防止制御信号変更部２６Ａは、ベクトル・レジスタ２１からのデータ供給を受けるリード処理期間と、リード処理期間後に結果をまとめ上げる収束期間とを必要とするベクトル命令については、該当ベクトル命令の収束処理を実行中の演算パイプライン２２－１，２２－２，…，２２－ｎに対する追越し防止制御を行なわないように、追越し防止制御部２５から出力される追越し防止制御信号を変更するものである。

【００６５】また、収束期間通知部２６Ｂは、各演算パイプライン２２－１，２２－２，…，２２－ｎが有する収束期間の開始信号(start final sequence;例えば収束期間開始時に立ち上がるパルス信号）と収束期間の終了信号(terminal final sequence；例えば収束期間終了時に立ち上がるパルス信号）とに基づいて、各演算パイプライン２２－１，２２－２，…，２２－ｎが前述した収束期間にあるか否かを、収束期間通知信号として対応する追越し防止制御信号変更部２６Ａへ出力するもので、例えば、図９に示すように構成されている。

【００６６】図９において、２６ａはＯＲゲート、２６ｂはＡＮＤゲート、２６ｃはフリップフロップ、２６ｅはインバータ（ＮＯＴゲート）であり、ＯＲゲート２６ａは、開始信号とフリップフロップ２６ｃからのデータ出力との論理和をとるものであり、ＡＮＤゲート２６ｂは、ＯＲゲート２６ａからの論理和出力と、終了信号のインバータ２６ｅによる反転信号との論理積をとるものである。

【００６７】また、フリップフロップ２６ｃは、ＡＮＤゲート２６ｂからの論理積出力に応じてデータ出力を

16

“１”に立ち上げるもので、ＯＲゲート２６ａに開始信号が入力されると、フリップフロップ２６ｃのデータ出力は立ち上がって“１”にセットされる一方、ＡＮＤゲート２６ｂに終了信号が入力されると、フリップフロップ２６ｃのデータ出力は“１”から“０”にリセットされるようになっている。つまり、フリップフロップ２６ｃのデータ出力（収束期間通知手段）は、各演算パイプライン２２－１，２２－２，…，２２－ｎが収束処理を実行中（収束期間中）、“１”に立ち上がっている。

【００６８】そして、追越し防止制御信号変更部２６Ａをなすゲート２６ｄは、追越し防止制御部２５により生成される追越し防止制御信号と、フリップフロップ２６ｃのデータ出力（収束期間通知手段）のインバータ（ＮＯＴゲート）２６ｆによる反転信号との論理積をとるもので、フリップフロップ２６ｃのデータ出力が“１”の間は、収束期間であるので、追越し防止制御信号は、ＡＮＤゲート２６ｄで強制的に“０”に変更される一方、フリップフロップ２６ｃのデータ出力が“０”の時は、追越し防止制御信号はそのまま出力されるようになっている。

【００６９】次に、上述のごとく構成されているベクトル処理装置の動作について、図１０～図１８を参照しながら説明する。

①〔該当命令の動作の説明〕

まず、該当命令の動作に関して説明する。ただし、実際のインプリメンテーションの都合により、ここに記す動作とは若干異なる動作をすることもある。ベクトル・レジスタ２１上のデータの総和を求める総和演算命令や、ベクトル・レジスタ２１上のデータの最大値もしくは最小値を求める検索演算命令では、元となるデータをベクトル・レジスタ２１から供給を受ける。

【００７０】総和演算命令では、順次、累和を求めることになるが、処理速度の向上のために、パイプラインのステージ分の部分累和を、多重にもつ演算器分だけ生成する。次に、総和演算の結果を求めるために、これらの部分和を全て足し込む。総和演算では、この部分和を足し込む処理を実行する期間が収束期間となる。検索演算命令では、順次、比較し選択することになるが、この場合も、ステージ数分の部分的な選択結果が、多重にもつ演算器分だけ生成された後、これらの部分的な選択結果の中から最終的に１つの結果を求める。検索演算では、この部分的な選択結果の中から最終的な１つの結果を求める操作を実行する期間が収束期間となる。

【００７１】これらの収束期間における処理は、ベクトル・レジスタ２１からのデータ供給を受けない処理のため、ベクトル・レジスタ２１のリンクによる追越し防止制御は不要である。

②〔一般的な追越し防止制御の説明〕

追越し防止制御を司るベクトル処理装置の部分は、一例をあげると次のようになっている。命令の発信と進行状

況とを管理する命令発信／管理部２７が命令を発信するときに、その命令の発信以前に実行を開始され、未だその実行を完了していない命令との間にオペランド・レジスタのリンクがあるかどうかをチェックし、リンクがあれば、追越し防止制御部２５にリンク条件が成立したことを通知する。

【００７２】このリンク条件は、リンクの成立したベクトル・レジスタ２１にデータを書き込む先行命令の実行を完了して、レジスタ・リンクが解消したときに解除される。レジスタ・リンクが複数存在するときは、全てのレジスタ・リンクが解消されたときにリンク条件を解除する。リンク条件の成立中は、先行命令のデータ書込が中断されることを検出すると、追越し防止制御部２５が起動される。この追越し防止制御部２５は、全ての演算パイプライン２２－１，２２－２，…，２２－ｎとストア・パイプライン２３Ｂとに対して処理を一時中断させるように通知する。ただし、リンクの起点（親）となるパイプラインについては、処理の中断は通知しない。先行命令のデータ書込を再開するか、もしくは、先行命令を完了すると、追越し防止制御部２５は、全ての演算パイプライン２２－１，２２－２，…，２２－ｎとストア・パイプライン２３Ｂとに対して処理を再開するように通知する。

【００７３】実際には、各演算パイプライン２２－１，２２－２，…，２２－ｎのベクトル・レジスタ２１からのデータ読出のスループットは、どの演算パイプライン２２－１，２２－２，…，２２－ｎでも等しいようにして、処理スループットの差による動的な追越し防止制御を不要としている。このため、スループットの変化するリンクの親となるパイプラインはロード・パイプライン２３Ｂに限定される。演算パイプライン２２－１，２２－２，…，２２－ｎ間の追越し防止については、全ての演算パイプライン２２－１，２２－２，…，２２－ｎに対して同時に処理の中断と再開とを指示することで回避されている。

【００７４】なお、インプリメンテーションの都合により、実際には、低スループットの演算パイプラインもベクトル処理装置上に存在している。このような場合には、通常は、この低スループットのパイプラインは、他のパイプラインとのリンク動作を禁止するように命令発信の段階で制御されている。とは言っても、より性能を追求する制御では、低スループットのパイプラインと言えども、制限付きでリンク動作をさせることがある。

【００７５】一例を挙げれば、先行する命令を処理するパイプラインのスループットが高いか等しければ、そのパイプラインの書込結果を読み出すような形のリンク動作を行なうことは可能である。禁止されているのは、低スループットの演算パイプラインの書込結果を使用する形で、高スループットのパイプラインが該低スループットのパイプラインにリンクして動作することである。

【００７６】③〔リード処理期間の説明〕

リード処理期間は、命令発信／管理部２７からの命令で最初に処理するデータを演算パイプライン２２－１，２２－２，…，２２－ｎがベクトル・レジスタ２１から受け取った時点を開始とし、その命令で最後に処理するデータを演算パイプライン２２－１，２２－２，…，２２－ｎがベクトル・レジスタ２１から受け取った時点を完了とする。

【００７７】④〔追越し防止制御信号変更部２６Ａ（ＡＮＤゲート２６ｄ）を追越し防止制御部２５側に、収束期間通知部２６Ｂを命令発信／管理部２７側にそなえた場合の説明〕

ここでは、命令発信／管理部２７から追越し防止制御部２５に対して、各演算パイプライン２２－１，２２－２，…，２２－ｎの処理が収束期間に入ったことを演算パイプライン情報とともに通知する機能と、収束期間が完了したことを演算パイプライン情報とともに通知する機能とが追加されている。つまり、図９により前述した収束期間通知部２６Ｂとしての機能が命令発信／管理部２７にそなえられている。

【００７８】また、追越し防止制御部２５には、各演算パイプライン２２－１，２２－２，…，２２－ｎに対する処理の一時中断（追越し防止制御信号）を通知する機能が各演算パイプライン２２－１，２２－２，…，２２－ｎ毎に設けられているが、その追越し防止制御部２５における各追越し防止制御信号出力用信号線毎に、追越し防止制御信号変更部２６Ａが設けられ、演算パイプライン２２－１，２２－２，…，２２－ｎのうち、収束期間にある演算パイプラインには処理の一時中断（追越し防止制御信号）が通知されないようになっている。

【００７９】具体的には、命令発信／管理部２７では、命令をデコードして収束期間をもつ命令であることを検出すると、その命令を発信する演算パイプライン２２－１，２２－２，…，２２－ｎの管理部（図示せず）に収束期間を有する命令である旨のフラグを立てる。演算パイプライン側管理部では、ベクトル長からリード期間に要する時間を計算し、追越し防止制御部２５からの処理の一時中断を通知する機能による情報を使用して、リード処理期間を正確に把握する。

【００８０】リード処理期間の完了とともに、収束期間が開始されるわけであるから、この時に追越し防止制御部２５に対して、収束期間通知部２６Ｂにより、その演算パイプライン２２－１，２２－２，…，２２－ｎが収束期間に入ったことを通知する。このとき通知される収束期間通知信号は、前述した通り、収束期間にあるときにはその演算パイプライン２２－１，２２－２，…，２２－ｎに対応する信号線が“１”に立ち上がり、それ以外のときは“０”となるものである。

【００８１】そして、追越し防止制御信号出力用の信号線上に、前述のように追越し防止制御信号変更部２６Ａ

をなすＡＮＤゲート２６ｄを設けることで、対応する演算パイプライン２２－１，２２－２，…，２２－ｎからの収束期間通知信号（フリップフロップ２６ｃのデータ出力）が“０”のときには、追越し防止制御信号は、そのままＡＮＤゲート２６ｄを通過して対応する演算パイプライン２２－１，２２－２，…，２２－ｎに通知される一方、対応する演算パイプライン２２－１，２２－２，…，２２－ｎからの収束期間通知信号が“１”のときには、追越し防止制御信号は、ＡＮＤゲート２６ｄにより“０”に変更されてから対応する演算パイプライン２２－１，２２－２，…，２２－ｎに通知される。

【００８２】これにより、従来、図１０に示すように、追越し防止制御信号の出力時には収束期間であっても完全に中断されていた処理が、本実施例では、収束期間中にある演算パイプラインについては変更された追越し防止制御信号を受けることにより、図１１に示すように、中断されることなく実行されるようになる。

⑤〔追越し防止制御信号変更部２６Ａおよび収束期間通知部２６Ｂを各演算パイプライン側にそなえた場合の動作の説明〕

ここでは、各演算パイプライン２２－１，２２－２，…，２２－ｎ毎に追越し防止制御信号変更部２６Ａおよび収束期間通知部２６Ｂをそなえ、各演算パイプライン自身のシーケンサ（開始信号，終了信号）に基づいて、収束期間通知部２６Ｂにより自分が収束期間にいることを検出し、収束期間中は、追越し防止制御信号変更部２６Ａにより、追越し防止制御部２５から各演算パイプライン２２－１，２２－２，…，２２－ｎに対する追越し防止制御信号を、“１”から“０”に変更するようにしている。

【００８３】なお、各演算パイプライン２２－１，２２－２，…，２２－ｎには、命令，ベクトル長，起動信号を受け取って処理を実行できるようにデータフローを制御するための内部シーケンサ（図示せず）が用意されている。この内部シーケンサは、データフローを制御するための命令の各種シーケンスを実行できるように、命令に伴うシーケンス連鎖の手順を滞りなく実行できるようになっている。

【００８４】収束期間通知部２６Ｂでは、リード処理期間の終了と収束期間開始との情報を内部シーケンサから受け取って、収束期間以外のときは、追越し防止制御信号変更部２６Ａ（ＡＮＤゲート２６ｄ）により、追越し防止制御信号をそのまま通過させるが、対応する演算パイプラインに対する収束期間のときには、追越し防止制御信号を“０”に変更する。

【００８５】これにより、上述のように追越し防止制御信号変更部２６Ａおよび収束期間通知部２６Ｂを各演算パイプライン側にそなえた場合にも、従来、図１０に示すように、追越し防止制御信号の出力時には収束期間であっても完全に中断されていた処理が、本実施例では、収束期間中にある演算パイプラインについては変更された追越し防止制御信号を受けることにより、図１１に示すように、中断されることなく実行されるようになる。

【００８６】⑥〔演算パイプラインの一構成例およびその動作の説明〕

ここで、演算パイプラインの一例として、図１３に示すように４段のステージから成るパイプライン式の加算器２２ａ～２２ｄを、図１２に示すように４個有する総和演算パイプライン２２Ａを示す。なお、図１３において、２８ａは第１ステージ・レジスタ、２８ｂは第２ステージ・レジスタ、２８ｃは第３ステージ・レジスタ、２８ｄは第４ステージ・レジスタ、２８ｅは転送用中継レジスタ、２９はセレクタ、３０は指数差計算手段、３１は桁合わせ手段、３２は加算／減算手段、３３は正規化手段である。また、各加算器２２ａ～２２ｄのステージ段数や、加算器の数については、処理装置毎に最適な段数が用意され、処理する命令（検索演算命令等）によっては加算器を比較選択手段とすることもできる。

【００８７】各加算器２２ａ～２２ｄの第１ステージは桁合わせに先立つ指数比較処理、第２ステージは桁合わせ処理を実行する。第３ステージは加算／減算処理を実行する。第４ステージは正規化処理である。加算器２２ａはベクトル・レジスタ中のデータ・エレメント番号の４による剰余が０のエレメントを処理し、加算器２２ｂはエレメント番号の４による剰余が１のエレメントを処理し、加算器２２ｃはエレメント番号の４による剰余が２のエレメントを処理し、加算器２２ｄはエレメント番号の４による剰余が３のエレメントを処理するように構成されている。

【００８８】また、各加算器２２ａ～２２ｄを一つにまとめるために、加算器２２ｂ～２２ｄから加算器２２ａに対して結果を転送することができるようになっている。総和演算では、ベクトル・レジスタ２１からのデータ供給を受けるリード処理期間中は、第４ステージのデータは第１ステージに戻されて、加算器２２ａ～２２ｄは累和加算器として動作する。また、４つの加算器２２ａ～２２ｄは並列に動作する。最終リード処理期間中は各加算器２２ａ～２２ｄ中に４つの部分和を生成する。

【００８９】加算器２２ａには、$\Sigma(E_{16i})$，$\Sigma(E_{16i+4})$，$\Sigma(E_{16i+8})$，$\Sigma(E_{16i+12})$の４種類の部分和が生成される。ここで、E_iはエレメント番号ｉのデータ・エレメントの値である。収束期間では、各部分和を取りまとめる手段は、幾つも存在しえるが、ここでは、その中の一例を図１４に示す。

【００９０】まず、各加算器２２ａ～２２ｄ中で、４つの部分和を足し合わせて各々１つの部分結果を生成する。次に、これらの４つの部分結果を足し合わせて最終的な結果を演算する。ここでも、加算器２２ａを例にとって、部分結果を生成する様子を説明する。収束期間に入った瞬間に、どの部分和がどのステージにいるかにつ

いては、ＶＬ長に依存するので、Σ（E_{16i}）の部分和が第１ステージ・レジスタ２８ａにセットされるまで空足しを行ない演算順序を保証する。第１ステージ・レジスタ２８ａで２τの間、部分和Σ（E_{16i}）をホールドする。

【００９１】その間に、第１ステージのもう一方のレジスタ２８ａに部分和Σ（E_{16i+4}）をセットする。部分和Σ（E_{16i}）と部分和Σ（E_{16i+4}）を第１ステージから第２ステージに移動させ加算処理を開始するとともに、部分和Σ（E_{16i+8}）を第１ステージ・レジスタ２８ａにセットし、２τの間、ホールドする。部分和Σ（E_{16i+8}）をホールド中に、もう一つの第１ステージ・レジスタ２８ａに部分和Σ（E_{16i+12}）をセットする。そして、部分和Σ（E_{16i+8}）と部分和Σ（E_{16i+12}）とを加算する。

【００９２】続いて、部分和｛Σ（E_{16i}）＋Σ（E_{16i+4}）｝を第１ステージ・レジスタ２８ａにセットし、３τの間、ホールドする。その間に部分和｛Σ（E_{16i+8}）＋Σ（E_{16i+12}）｝をもう一方の第１ステージ・レジスタ２８ａにセットして加算処理を実行して加算器２２ａの中間結果を求める。収束期間における、この中間和を求める動作は、各加算器２２ａ～２２ｄで並行して行なわれる。

【００９３】最終的な結果を求めるための演算は、加算器２２ａで実行する。加算器２２ａでは、第１ステージ・レジスタ２８ａに加算器２２ａの中間和をホールドし、加算器２２ｂの中間和がもう一方の第１ステージ・レジスタ２８ａにセットされるのを待つ。そして、加算器２２ａと加算器２２ｂとの中間和どうしを加算して中間和Ａを求める。この中間和Ａは、第１ステージ・レジスタ２８ａ上でホールドされ、加算器２２ｃの中間和がもう一方の第１ステージ・レジスタ２８ａにセットされるのを待つ。そして、中間和Ａと加算器２２ｃの中間和が加算されて、中間和Ｂが生成される。この中間和Ｂは、第１ステージ・レジスタ２８ａでホールドされ、加算器２２ｄの中間和がもう一方の第１ステージ・レジスタ２８ａにセットされるのを待つ。そして、中間和Ｂと加算器２２ｄの中間和とを加算して、最終的な結果を生成する。

【００９４】⑦〔付加演算器付き演算パイプラインに本発明を適用した実施例の説明〕

次に、各演算パイプライン２２－１，２２－２，…，２２－ｎが、基本演算器と収束を処理する付加演算器とを有する構成のもので（例えば図１７参照）、収束処理を付加演算器により実行し、収束処理中、基本演算器により後続の他の演算命令を実行できるものである場合について説明する。

【００９５】このような場合も、基本演算器から切り離されて収束期間に入ってから、その命令を完了するまでの収束期間中、付加演算器では、追越し防止制御部２５

から各演算パイプライン２２－１，２２－２，…，２２－ｎに対する処理の一時中断を通知する手段による情報（追越し防止制御信号）を、追越し防止制御信号変更部２６Ａにより受け付けないようにしている。

【００９６】付加演算器を使用する場合には、内部シーケンサについても、基本演算器部分と付加演算器部分とに分割されている。付加演算器部分の内部シーケンサは、基本演算器部分の内部シーケンサから起動され、収束期間に入る時点で基本演算器部分から切り離される旨の通知を受ける。付加演算器部分の内部シーケンサは、基本演算器部分から切り離されると収束のためのシーケンスを起動する。

【００９７】収束期間通知部２６Ｂは、リード期間の終了と収束期間開始との情報を付加演算器部分の内部シーケンサから受け取って、追越し防止制御信号変更部２６Ａにより、収束期間以外のときは、追越し防止制御信号をそのまま通過させるが、対応する演算パイプライン２２－１，２２－２，…，２２－ｎの付加演算器に対する収束期間の時には追越し防止制御信号を“０”に変更する。

【００９８】基本演算器部分については、付加演算器部分を切り離した後は、付加演算器部分が演算を完了するまで、付加演算器部分を必要としない演算を実行可能として、命令発信／管理部２７からの起動待ちとなる。付加演算器部分が命令を完了し且つ基本演算器部分が付加演算器部分を使用しない命令のリード処理期間を完了するか、付加演算器部分と基本演算器部分との双方ともに命令の実行を完了していれば、その演算パイプライン２２－１，２２－２，…，２２－ｎは、付加演算器を使用する命令を実行可能になる。

【００９９】これにより、収束処理を行なう付加演算器付きの演算パイプラインについても、従来、図１５に示すように、追越し防止制御信号の出力時には収束期間であっても完全に中断されていた処理が、本実施例では、収束期間中にある演算パイプラインの付加演算器については変更された追越し防止制御信号を受けることにより、図１６に示すように、中断されることなく実行されるようになる。

【０１００】⑧〔付加演算器付き演算パイプラインの一構成例およびその動作の説明〕

ここでは、付加演算器付き演算パイプラインの一例として、図１７に示すように、４段のステージから成るパイプライン式の基本演算器（図１３に示した加算器と同様構成のもの）３４ａ～３４ｄと、４段のステージから成るパイプライン式の付加演算器（図１３に示した加算器と同様構成のもの）３５ａ～３５ｄとを組み合わせた複合演算器３６ａ～３６ｄを４つもつような総和演算パイプライン２２Ｂを示す。

【０１０１】なお、ステージ段数や加算器の数については、処理装置毎に最適な段数が用意され、処理する命令

（検索演算命令等）によっては加算器を比較選択手段とすることもできる。また、前述したものと同様に、各演算器（加算器）３４ａ～３４ｄおよび３５ａ～３５ｄの第１ステージは桁合わせに先立つ指数比較処理、第２ステージは桁合わせ処理を実行する。第３ステージは加算／減算処理を実行する。第４ステージは正規化処理である。

【０１０２】複合演算器３６ａはベクトル・レジスタ２１中のデータ・エレメント番号の８による剰余が０と１のエレメントを処理し、複合演算器３６ｂはエレメント番号の８による剰余が２と３のエレメントを処理し、複合演算器３６ｃはエレメント番号の８による剰余が４と５のエレメントを処理し、複合演算器３６ｄはエレメント番号の８による剰余が６と７のエレメントを処理するように構成されている。

【０１０３】また、各付加加算器３５ａ～３５ｄの結果を一つにまとめるために、付加加算器３５ｂ，付加加算器３５ｃ，付加加算器３５ｄから付加加算器３５ａに対して結果を転送できるようになっている。そして、リード処理期間には、基本演算器３４ａ～３４ｄからなる基本演算部３４では、連続する２つのエレメントの和を求めて、付加演算器３５ａ～３５ｄからなる付加演算部３５に転送する。付加演算部３５は基本演算部３４から受け取った中間和を足し込む。リード期間が終了するときには、付加演算部内に４つの部分和が生成されている。複合演算器３６ａを例にとると、この４つの部分和は、

$\Sigma(E_{32i}+E_{32i+1})$，$\Sigma(E_{32i+8}+E_{32i+9})$，

$\Sigma(E_{32i+16}+E_{32i+17})$，$\Sigma(E_{32i+24}+E_{32i+25})$

となる。ここで、E_iはエレメント番号ｉのデータ・エレメントの値である。

【０１０４】収束期間に入ると、付加演算部３５は基本演算部３４のデータバスからのデータを受け付けないようにする。続いて、４つの部分和から中間結果を計算し、最後に４つの中間結果から最終結果を計算する。複合演算器３６ａを例にとって中間結果を求める手順を図１８により説明する。

【０１０５】まず、演算順序を保証するために、部分和$\Sigma(E_{32i}+E_{32i+1})$が第１ステージ・レジスタ２８ａ上に来るまで空足しを行なう。その部分和は第１ステージ・レジスタ２８ａ上で２τ間ホールドされる。その間に部分和$\Sigma(E_{32i+8}+E_{32i+9})$をもう一つの第１ステージ・レジスタ２８ａにセットする。そして、部分和Ａの計算を開始する。

【０１０６】次に、部分和$\Sigma(E_{32i+16}+E_{32i+17})$を第１ステージ・レジスタ２８ａにセットし２τの間ホールドする。その間に部分和$\Sigma(E_{32i+24}+E_{32i+25})$をもう一方の第１ステージ・レジスタ２８ａにセットする。そして、もう一つの部分和Ｂを計算する。部分和Ａを第１ステージ・レジスタ２８ａに２τの間ホールドし、その間に部分和Ｂをもう一方の第１ステージ・レジスタ２８ａにセットし、そして中間結果を計算する。

【０１０７】収束期間における、この中間結果を求める動作は、各付加演算器３５ａ～３５ｄで並行して行なわれる。最終的な結果を求めるための演算は、付加演算器３５ａで実行する。付加演算器３５ａでは、第１ステージ・レジスタ２８ａに付加演算器３５ａの中間結果をホールドし、付加演算器３５ｂの中間結果がもう一方の第１ステージ・レジスタ２８ａにセットされるのを待つ。そして、付加演算器３５ａと付加演算器３５ｂの中間結果どうしを加算して中間結果Ａを求める。この中間結果Ａは、第１ステージ・レジスタ２８ａ上でホールドされ、付加演算器３５ｃの中間結果がもう一方の第１ステージ・レジスタ２８ａにセットされるのを待つ。

【０１０８】そして、中間結果Ａと付加演算器３５ｃの中間結果とが加算されて中間結果Ｂが生成される。この中間結果Ｂは、第１ステージ・レジスタ２８ａでホールドされ、付加演算器３５ｄの中間結果がもう一方の第１ステージ・レジスタ２８ａにセットされるのを待つ。そして、中間結果Ｂと付加演算器３５ｄの中間結果とを加算して、最終的な結果を生成する。

【０１０９】このように、本発明の第２実施例のベクトル処理装置によれば、追越し防止制御部２５から追越し防止制御信号が出力された際に、演算パイプライン２２－１，２２－２，…，２２－ｎ，２２Ａ，２２Ｂが収束処理のシーケンスを実行中で、収束期間条件が成立している間は、追越し防止制御信号変更部２６Ａにより追越し防止制御部２５からの追越し防止制御信号が変更され、収束処理中の演算パイプライン２２に対する追越し防止制御が禁止されるので、追越し防止制御信号変更部２６Ａおよび収束期間通知部２６Ｂという極めて少量の物量の増加によるだけで、収束期間におけるレジスタ・リンクによる追越し制御のオーバヘッドを回避することができ、性能の向上を実現することができる。

【０１１０】

【発明の効果】以上詳述したように、本発明のベクトル処理装置（請求項１，２）によれば、スループットの少ない演算パイプラインを、１つのバンクスロットしか使用しないメモリ・アクセス・パイプライン（ストア・パイプライン）と共用し、演算パイプラインをオーバラップさせて実行することにより、演算スループットの大幅な向上を実現できる効果がある。

【０１１１】また、本発明のベクトル処理装置（請求項３～５）によれば、追越し防止制御部２５から追越し防止制御信号が出力された際に、演算パイプラインが収束処理のシーケンスを実行中で、収束期間条件が成立している間は、変更部追越し防止制御信号が変更され、収束処理中の演算パイプラインに対する追越し防止制御が禁止されるので、処理速度の大幅な改善を実現できる効果がある。

【図面の簡単な説明】

25

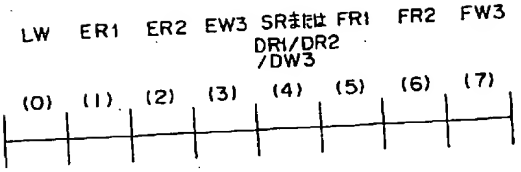
【図１】第１の発明の原理説明図である。
【図２】第２の発明の原理ブロック図である。
【図３】本発明の第１実施例としてのベクトル処理装置を示すブロック図である。
【図４】第１実施例の動作を説明するためのタイミングチャートである。
【図５】第１実施例のバンク管理部の構成例を示すブロック図である。
【図６】第１実施例の動作を説明するためのタイミングチャートである。
【図７】第１実施例の動作を説明するためのタイミングチャートである。
【図８】本発明の第２実施例としてのベクトル処理装置を示すブロック図である。
【図９】第２実施例における追越し防止制御信号変更部の構成を示す回路図である。
【図１０】第２実施例の動作を説明するための図である。
【図１１】第２実施例の動作を説明するための図である。
【図１２】第２実施例の演算パイプラインの構成例を示すブロック図である。
【図１３】第２実施例の加算器の構成例を示すブロック図である。
【図１４】第２実施例の演算パイプラインの動作例を説明するためのタイミングチャートである。
【図１５】第２実施例の動作を説明するための図である。
【図１６】第２実施例の動作を説明するための図である。
【図１７】第２実施例の付加演算器付き演算パイプラインの構成例を示すブロック図である。
【図１８】第２実施例の付加演算器付き演算パイプラインの動作例を説明するためのタイミングチャートである。
【図１９】一般的なバンクスロットのタイミング設定例を示すタイミングチャートである。
【符号の説明】
１－０，１－１，…，１－ｎ　ベクトル・レジスタ
２　メモリ・アクセス・パイプライン
２Ａ　ロード・パイプライン
２Ｂ　ストア・パイプライン
３Ａ，３Ｂ－０，３Ｂ－１，…，３Ｂ－ｍ　書込レジスタ
４Ａ，４Ｂ－０，４Ｃ－０，４Ｂ－１，４Ｃ－１，…，４Ｂ－ｍ，４Ｃ－ｍ読出レジスタ

26

＊５－０，５－１，…，５－ｍ　演算パイプライン
６　命令制御部
７　バンク管理部
７ａ　バンクスロット・カウンタ
１１－１，１１－２，１１－３　管理レジスタ
１２　バンクスロット割当回路
１３　通知レジスタ
１４　起動信号制御部
１５　割算パイプライン
１５ａ～１５ｆ　割算器
２０　主記憶部
２１　ベクトル・レジスタ
２２，２２－１，２２－２，…，２２－ｎ　演算パイプライン
２２Ａ，２２Ｂ　総和演算パイプライン
２２ａ～２２ｄ　加算器
２３－１，２３－２，…，２３－ｍ　メモリ・アクセス・パイプライン
２３，２３Ａ　ロード・パイプライン
２３Ｂ　ストア・パイプライン
２４　主記憶部
２５　追越し防止制御部
２６　変更部
２６Ａ　追越し防止制御信号変更部
２６Ｂ　収束期間通知部
２６ａ　ＯＲゲート
２６ｂ　ＡＮＤゲート
２６ｃ　フリップフロップ
２６ｄ　ＡＮＤゲート
２６ｅ，２６ｆ　インバータ（ＮＯＴゲート）
２７　命令発信／管理部
２８ａ　第１ステージ・レジスタ
２８ｂ　第２ステージ・レジスタ
２８ｃ　第３ステージ・レジスタ
２８ｄ　第４ステージ・レジスタ
２８ｅ　転送用中継レジスタ
２９　セレクタ
３０　指数差計算手段
３１　桁合わせ手段
３２　加算／減算手段
３３　正規化手段
３４　基本演算部
３４ａ～３４ｄ　基本演算器
３５　付加演算部
３５ａ～３５ｄ　付加演算器
３６ａ～３６ｄ　複合演算器

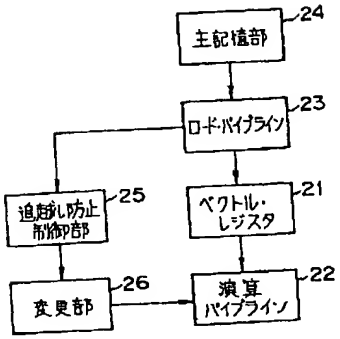
【図１】

第１の発明の原理説明図



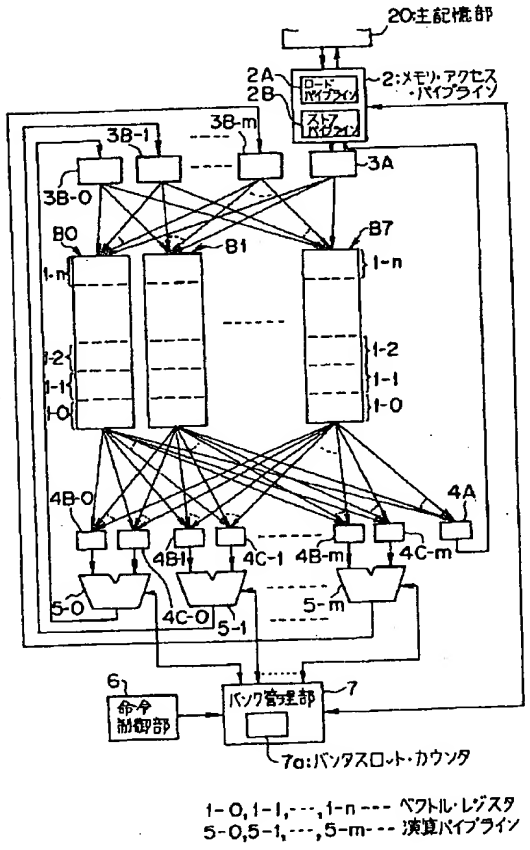
【図２】

第２の発明の原理ブロック図



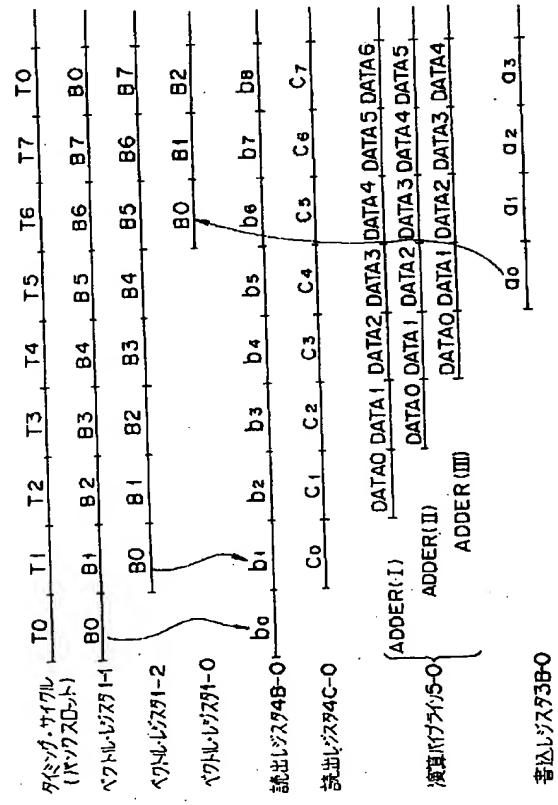
【図３】

本発明の第１実施例としてのベクトル処理装置を示すブロック図



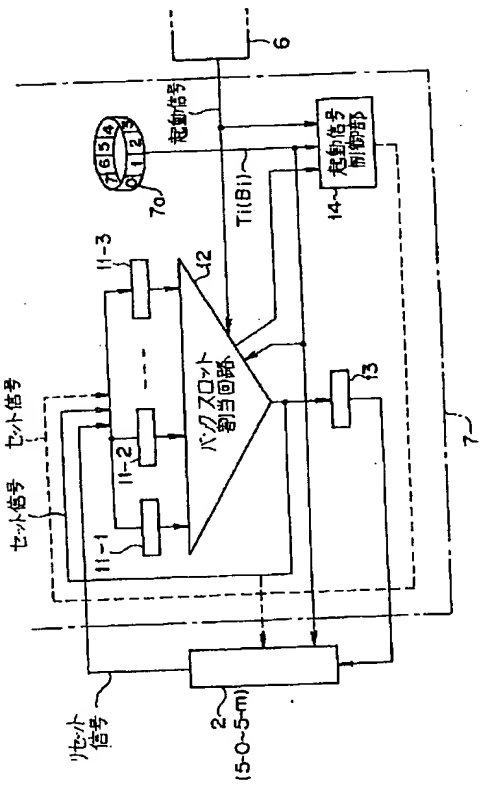
【図４】

第１実施例の動作を説明するためのタイミングチャート



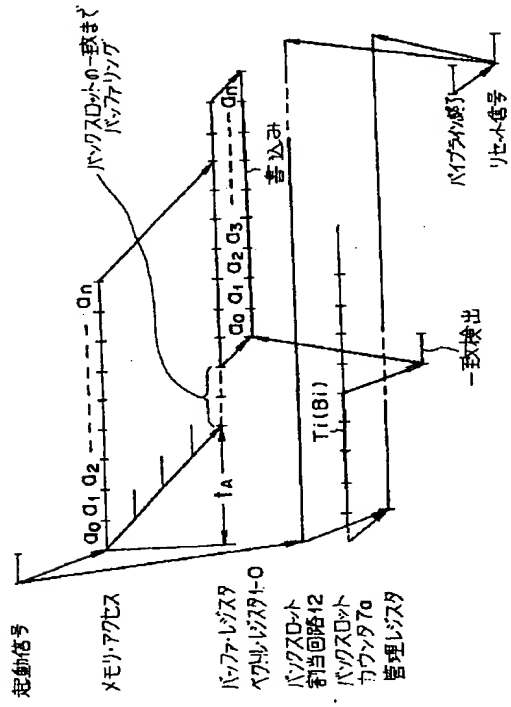
【図５】

第1実施例のバンク管理部の構成例を示すブロック図



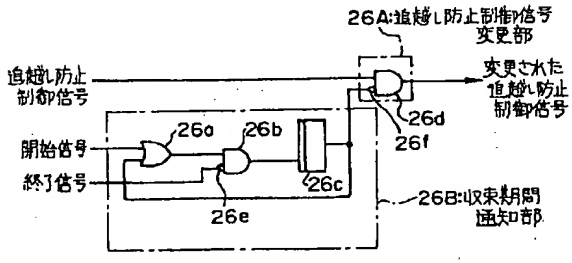
【図６】

第1実施例の動作を説明するためのタイミングチャート



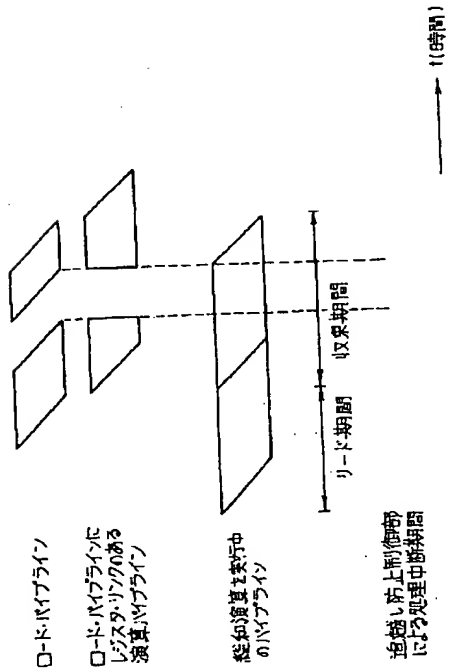
【図９】

第2実施例における追越し防止制御信号変更部の構成を示す回路図

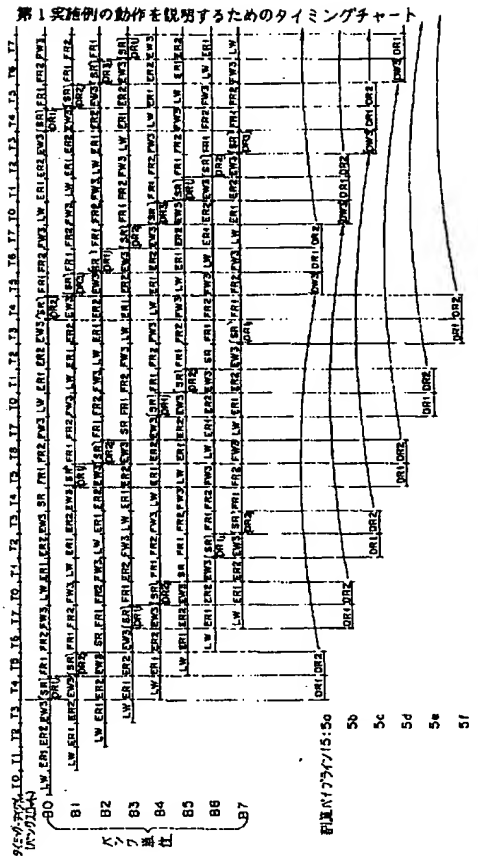


【図１１】

第2実施例の動作を説明するための図

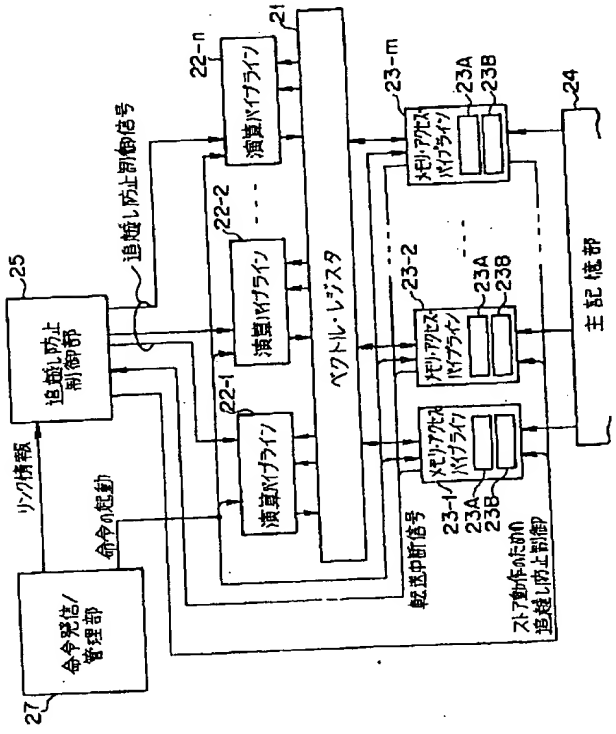


【図７】



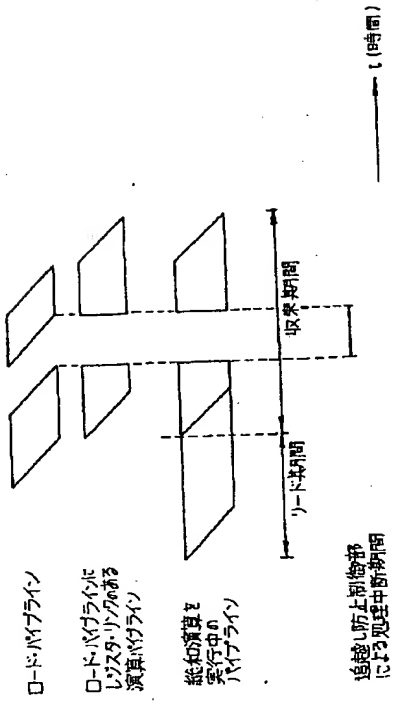
【図８】

本発明の第２実施例としてのベクトル処理装置を示すブロック図



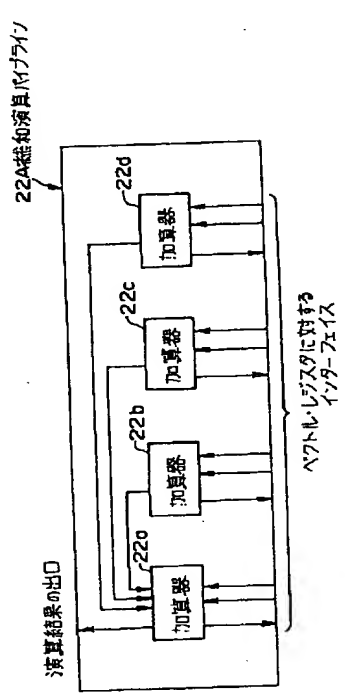
【図１０】

第2実施例の動作を説明するための図



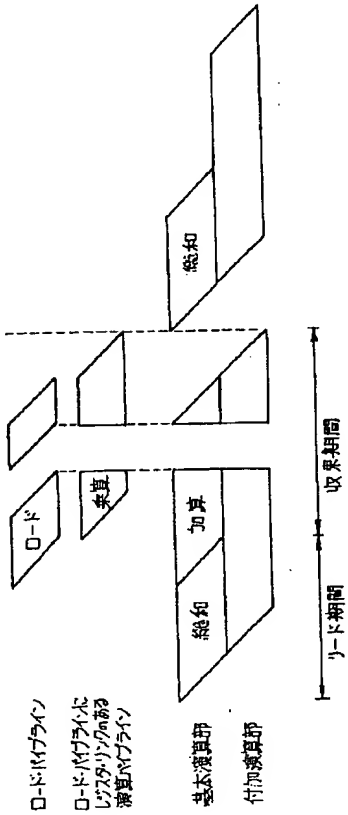
【図１２】

第2実施例の演算パイプラインの構成例を示すブロック図



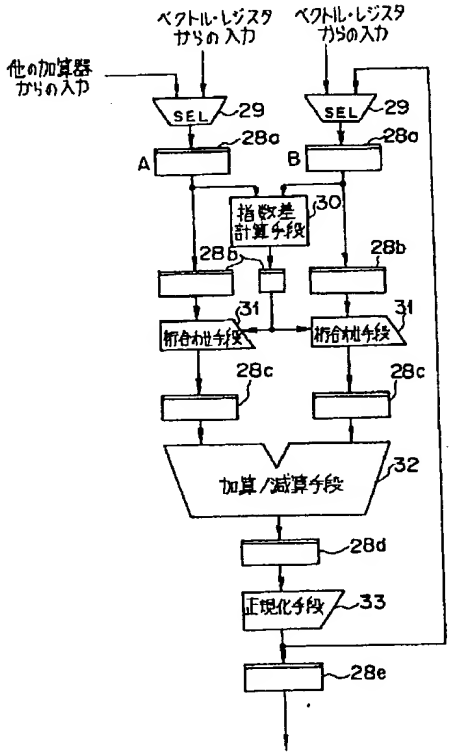
【図１５】

第2実施例の動作を説明するための図



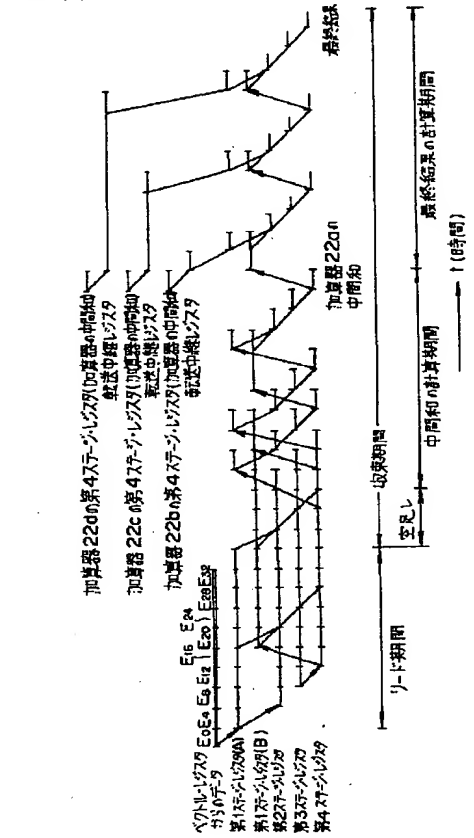
【図１３】

第2実施例の加算器の構成例を示すブロック図



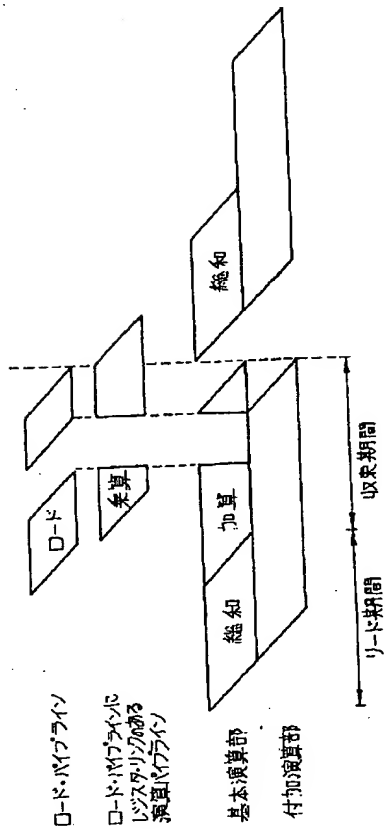
【図１４】

第2実施例の演算パイプラインの動作例を説明するためのタイミングチャート



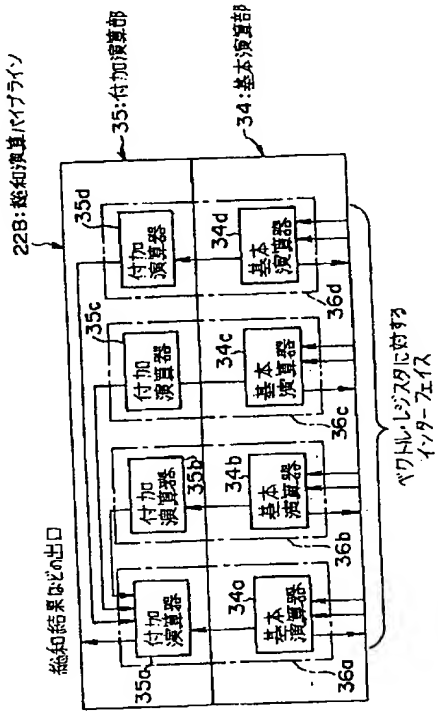
【図１６】

第２実施例の動作を説明するための図

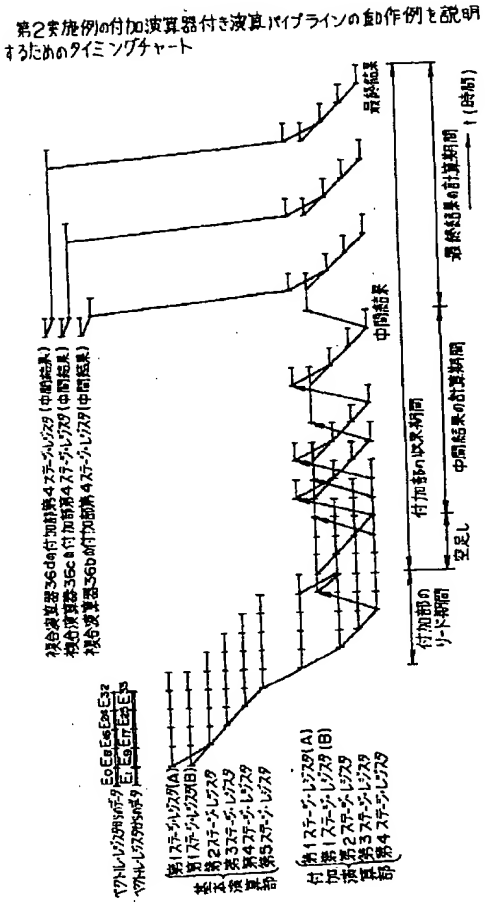


【図１７】

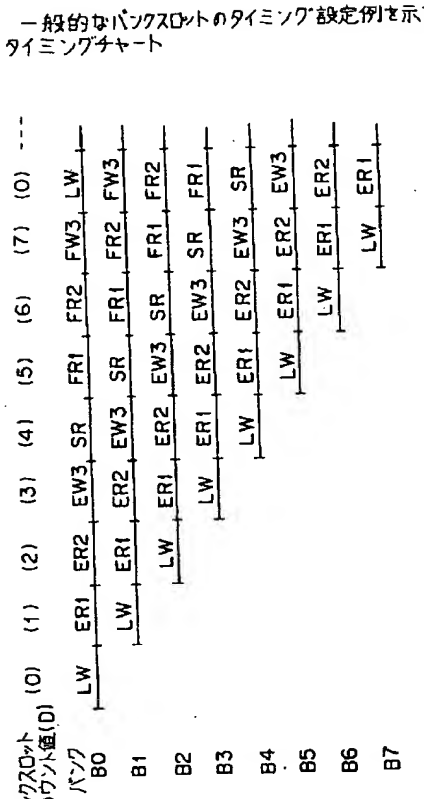
第２実施例の付加演算器付き演算パイプラインの構成例を示すブロック図



【図１８】



【図１９】



フロントページの続き

(72)発明者　今野　勝彦
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　渥美　宏昭
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内